IN THE UNITED STATES DISTRICT COURT
EASTERN DISTRICT OF TEXAS
TYLER DIVISION

THE PACID GROUP, LLC, Plaintiff, v. APPLE, INC. et al, Defendants.	Case No. 6:09-CV-143 (LED) (JDL)

DEFENDANTS' CLAIM CONSTRUCTION BRIEF

TABLE OF CONTENTS

TABLE OF AUTHORITIES

Page(s)

CASES

I. INTRODUCTION

This case concerns two patents that the PACid Group, LLC ("PACid") is attempting to stretch to cover a WiFi industry standard, known as IEEE 802.11i. Although the WiFi Alliance adopted the allegedly infringing portions of the IEEE 802.11i standard in 2002 and the patents-in-suit issued in 1999 and 2000, PACid waited until 2009 to file suit against the WiFi chip suppliers in this case—Atheros Communications, Inc., Broadcom Corporation, Intel Corporation, and Marvell Semiconductor Inc. (collectively, "Defendants"). This long delay despite the publicity surrounding the IEEE 802.11i standard speaks volumes about PACid's efforts to reinterpret the claims so that they allegedly cover the standard.

As is common in patent cases, the key issues are ones of claim construction. Defendants offer constructions grounded in the language of the claims and the definitional guidance that the intrinsic evidence provides. PACid, on the other hand, runs from the definitions set forth in the patents and the unbiased reference materials cited in those patents. This Court should adopt Defendants' proposed constructions because they are based on the application of well-entrenched canons of claim construction, as evidenced by the intrinsic and extrinsic evidence of record.

II. PACID'S PATENTS-IN-SUIT

The technology at issue in this case relates to generating symmetric encryption keys that are used to encrypt and decrypt information files on a computer. ('646 Patent, col. 1:17-43, Ex. 1; '612 Patent, col. 1:8-14, 27-49, Ex. 2.) Encryption transforms information (referred to as plaintext) using an algorithm (called a cipher) to make an output that is unintelligible (called ciphertext) to anyone except those possessing a special piece of information, usually referred to as a key. (Schneier, Bruce, Applied Cryptography 1 (2d ed. 1996) (hereinafter "Schneier Book"), Ex. 3.) Decryption is the reverse of encryption—namely, converting the unintelligible

ciphertext back to plaintext using the decryption key. (Id. at 1.) Symmetric key encryption refers to encryption methods in which the same key encrypts and decrypts the information. (Id. at 4; see also '646 Patent, col. 1:51-55; '612 Patent, col. 1:58-62.)

A. The '646 Patent

U.S. Patent No. 5,963,646 ("'646 patent") relates to a method and system for generating deterministic, symmetric encryption keys used to secure data in computer systems. ('646 Patent, col. 1:17- 21; col. 1:35-62; col. 3:1-23.) The patent suggests that prior art key generators were susceptible to attack that would reveal the inputs to the key generator. (Id. at col. 2:21-29.) To address this issue, the patent proposes a key generator that uses a secure hash to generate symmetric keys. (Id. at col. 3:1-6; col. 3:19-30; col. 4:55–col. 5:5.) Notably, the patent acknowledges that prior art key generators also incorporated secure hash functions. (Id. at col. 2:26-29.)

Figure 2 illustrates that the patent's key generation method includes a bit-shuffling generator and a secure hash generator. (Id. at col. 3:1-6; col. 4:55–col. 5:5; Fig. 2.). In particular:

- A secure E-Key Seed and a constant value are inputs to the bit-shuffling generator which mixes and maps these inputs to output a result of fewer total bits that the combination of the E-Key Seed and the constant value. (Id.)
- The bit shuffle result is inputted into the secure hash generator which in turn produces a pseudo-random bit sequence called a message digest. (Id.)
- The message digest may be truncated to a desired length to produce the deterministic encryption key. (Id. at col. 5:11-13.)

B. The '612 Patent

U.S. Patent No. 6,049,612 ("'612 patent") relates to a method and system for protecting sensitive information files and messages from access by unauthorized parties. ('612 Patent, col. 1:8-14.) The '612 patent relates to encrypting an information file with the key of the '646 patent

and undertaking additional security measures. (See '646 Patent, Fig. 2; '612 Patent, Figs. 3, 4.)

Specifically, Figures 3 and 4 of the '612 patent show that:

- The file security system combines a constant value and a secret E-Key Seed in a bit shuffling generator to shuffle bits and perform a first many-to-few bit mapping to provide a first pseudo-random result. (Fig. 3.)
- A secure one-way hash algorithm receives the first result and performs a second many-to-few bit mapping to produce a pseudo-random message digest. (Id.; '612 Patent, col. 3:17-33; col. 4:46-63.)
- The message digest may be truncated to provide a deterministic encryption key. (Id. at col. 5:18-20.)
- The security system subsequently encrypts the information file to be protected. (Id. at col. 5:37-42.)
- To detect any alternations to the encrypted information file, the system concatenates the constant value to the header of the encrypted information file and applies a secure hash to the combination to create a message integrity code. (Id.; col. 2:54–col. 3:9.) Any alteration of the encrypted file is reflected by the message integrity code. (Id. at col. 3:29-33.)

III. PROPOSED CLAIM CONSTRUCTIONS

Because this Court is well aware of the developed jurisprudence governing claim construction, this brief will not repeat it herein.

A. The '646 Patent

1. "pseudo-random"[1]

Defendants Proposed Construction	PACid Proposed Construction
"refers to output that is repeatable and predictable to anyone who knows the function's input but appears to be totally random to those without such knowledge"	No construction necessary; alternatively: "apparently random, but repeatable and predictable"

As an initial matter, PACid would have this Court provide no instruction to the jury regarding the meaning of the technical term "pseudo-random," despite the fact that the patentee explicitly defined this term in the specifications of both patents. ('646 Patent, col. 5:14-18; '612

[1] "Pseudo-random" appears in claims 1, 12, 17 and 26 of the '646 patent and claims 1 and 11 of the '612 patent. The parties agree that this term has the same meaning in all of the asserted claims.

Patent, col. 4:1-5.) Unquestionably, a lay juror may not be familiar with the term "pseudo-random" either generally or in the specific context of the PACid patents. Accordingly, this Court should construe it. See Embrex, Inc. v. Serv. Eng'g Corp., 216 F.3d 1343, 1347 (Fed. Cir. 2000) ("The construction of claims is simply a way of elaborating the normally terse claim language in order to understand and explain, but not to change, the scope of the claims." (internal quotation and citation omitted)).

Indeed, to ensure that those of ordinary skill in the art would understand the meaning of this term, the named inventors expressly defined it in clear, user-friendly terms:

> The term "pseudo-random" as used in this specification means that the output referred to is repeatable and predictable to anyone who knows the E-Key seed input to the function producing the output. Without such knowledge, the output appears to be totally random.

('646 Patent, col. 5:14-18; '612 Patent, col. 4:1-5.) "When a patentee explicitly defines a claim term in the patent specification, the patentee's definition controls." Martek Biosciences Corp. v. Nutrinova, Inc., 579 F.3d 1363, 1380 (Fed. Cir. 2009).

Nonetheless, PACid retreats from the patents' explicit definition, and instead offers a construction that cherry-picks portions of the definition while discarding several of its key aspects. PACid's construction eliminates three key requirements: (1) "without knowledge of the function's input," the output "appears to be totally random"; (2) the function is repeatable and predictable "to anyone who knows the function's input"; and (3) the output appears to be "totally" random. These deletions incorrectly broaden the term beyond its express definition. For example, in the patents' definition of "pseudo-random," the adjective "totally" modifies the word "random," and makes clear that the entire result appears totally random.

Before the patents even issued, the inventors defined "pseudo-random" to include these three requirements. PACid's attempts to retreat from these limitations years later in litigation are

improper. Accordingly, this Court should reject PACid's attempt to rewrite the definition of "pseudo-random."

2. "secure hash operation," "secure hash algorithm," "secure hash computer program"

Claim Limitation	Defendants Proposed Construction	PACid Proposed Construction
"secure hash operation" ('646 claim 1) ('612 claim 1)	"an operation that accepts an input that can be of variable bit length, but always produces an output having the same bit length such that it is computationally infeasible to determine (a) the input from the output and (b) two inputs that produce the same output, and where if a single bit of the input is changed, on average approximately 50% of the output bits are changed"	"algorithm that produces a deterministic output having no known relationship with the input that may be used to recover the input from the output"
"secure hash algorithm" ('646 claim 12)	"an algorithm that accepts an input that can be of variable bit length, but always produces an output having the same bit length such that it is computationally infeasible to determine (a) the input from the output and (b) two inputs that produce the same output, and where if a single bit of the input is changed, on average approximately 50% of the output bits are changed"	"algorithm that produces a deterministic output having no known relationship with the input that may be used to recover input from the output"
"secure hash computer program" ('646 claim 12)	"computer program that uses a secure hash algorithm (as defined above)"	No separate construction (in light of the others); alternatively: "A computer program that produces a deterministic output having no known relationship with the input that may be used to recover the input from the output."

For all three of these terms, the dispute centers on the meaning of "secure hash" and PACid's efforts to distance itself from the standard definition of "secure hash,"[2] even though the PACid patents embrace this standard definition throughout their specifications. This well-known definition of "secure hash" is also consistent with other intrinsic evidence, including the

[2] A secure hash is a well-known tool used in cryptography. (See, e.g., Schneier Book at 429-459 (chapter on one-way hash functions), Ex. 3; see also Menezes, Alfred J. et al., Handbook of Applied Cryptography 321-383 (1996) (hereinafter "Menezes Book") (chapter on cryptographic hash functions), Ex. 4; id. at 321 ("Cryptographic hash functions play a fundamental role in modern cryptography.").)

technical standards and the prior art cited and discussed in the PACid patents.[3] Defendants' proposed constructions include the four crucial requirements of a secure hash that the inventors embraced in the patents and the cited prior art confirms. In contrast, PACid's constructions improperly omit at least three of these requirements.

(a) The first key property of a "secure hash" function is that the "output is always the same binary length regardless of the size of the input"

First, the patent specification teaches that a "characteristic of a hash algorithm is that the output is always the same binary length regardless of the size of the input." ('646 Patent, col. 2:2-4 (emphasis added); see also id. at col. 5:5-11; '612 Patent, col. 2:9-13.) The '612 patent confirms that this feature is an integral one of the invention and not merely a preferred embodiment. Specifically, the '612 patent provides that, in "the present invention," the "key generation method which is used employs . . . a secure hash function which produces a message digest of constant binary length (no matter the binary length of the input)." ('612 Patent, col. 2:54-65.) See nCube Corp. v. Seachange Int'l, Inc., 436 F.3d 1317, 1329 (Fed. Cir. 2006) ("[T]he use of the term 'present invention' is strong evidence that the use of logical addressing applies to the invention as a whole, not just the preferred embodiment.").

This variable bit input/same output length is a well-known property of a secure hash algorithm as the cited prior art references confirm. For example, both patents disclose the Secure Hash Standard, FIPS PUB 180-1 (also called SHA-1) as an example of a "secure hash." ('646 Patent, col. 2:26-28, col. 5:27-28; '612 Patent, col. 2:2-5, col. 5:2-4). SHA-1 accepts an input that can be of variable bit length, but always produces an output having the same bit length.

[3] "[P]rior art cited in a patent or cited in the prosecution history of the patent constitutes intrinsic evidence." V-Formation, Inc. v. Benetton Group SPA, 401 F.3d 1307, 1311 (Fed. Cir. 2005); see also Arthur A. Collins, Inc. v. N. Telecom Ltd., 216 F.3d 1042, 1045 (Fed. Cir. 2000) (Cited prior art "can have particular value as a guide to the proper construction of the term, because it may indicate not only the meaning of the term to persons skilled in the art, but also that the patentee intended to adopt that meaning.").

(FIPS 180-1 at DEFS0006337 ("When a message of any length < 2^{64} bits is input, the SHA-1 produces a 160-bit output called a message digest."), Ex. 5.)

Likewise, standard texts, including the Schneier Book cited in the '612 patent,[4] explain that a "secure hash" function accepts an input of variable bit length, but always produces an output having the same bit length. The Schneier Book explains: "A one-way hash function,[5] H(M), operates on an arbitrary-length pre-image message M [and] returns a fixed-length hash value, h." (Schneier Book at 429, Ex. 3). Other prior art technical references are in accord. (See, e.g., Menezes Book at 321 ("[A] [cryptographic] hash function h maps bit-strings of arbitrary finite lengths to strings of fixed lengths."), Ex. 4;[6] see also id. at 322 ("[Cryptographic hash function] h maps an input x of arbitrary finite bit length, to an output h(x) of fixed bit length.").) Accordingly, both the intrinsic evidence and unbiased technical references from the time of the PACid patents' effective filing date (1997) uniformly show that a key requirement of a "secure hash" function is that it accepts an input that can be of variable bit length, but always produces an output having the same bit length.

[4] The '612 patent discloses the Schneier Book and instructs those of ordinary skill in the art to consult this reference for "general information related to file encryption techniques." ('612 Patent, col. 2:49-51.) Because the Schneier Book is cited prior art, it is intrinsic evidence. V-Formation, 401 F.3d at 1311 ("[P]rior art cited in a patent or cited in the prosecution history of the patent constitutes intrinsic evidence.").

[5] The PACid patents use secure hash and one-way hash interchangeably. (See '646 Patent, col. 2:26-36 ("In order to introduce a higher degree of irreversibility, secure one-way hash functions such as that defined in 'Secure Hash Standard', FIPS PUB 180-1 (Apr. 17. 1995), have been introduced into the key generation process.").)

[6] Although the Menezes Book is not intrinsic evidence, a court may consider "trustworthy" extrinsic evidence to ensure that its claim construction is not inconsistent with "clearly expressed, plainly apposite, and widely held understandings in the pertinent technical field." Pitney Bowes, Inc. v. Hewlett-Packard Co., 182 F.3d 1298, 1309 (Fed. Cir. 1999). This check is especially useful for technical terms. Phillips v. AWH Corp., 415 F.3d 1303, 1317 (Fed. Cir. 2005) (en banc). Such extrinsic evidence may take the form of expert testimony, dictionaries, technical treatises, and articles. Id. Courts may not rely, however, on extrinsic evidence to contradict or vary the meaning of claims provided by the intrinsic evidence. Id. at 1318.

(b) The second key property of a "secure hash" function is that it is computationally infeasible to determine the input from the output

The second key property of a "secure hash" function is that it is computationally infeasible to determine the input from the output. The written descriptions of the PACid patents confirm this requirement when they:

- Explain that "[t]here is no known relationship between the input and output of a hash algorithm which may be used to recover the input from the output." ('646 Patent, col. 2:8-10 (emphasis added); see also '612 Patent, col. 2:15-17.)

- Teach that a secure hash algorithm is irreversible. ('646 Patent, col. 2:26-36 ("In order to introduce a higher degree of irreversibility, secure one-way hash functions such as that defined in 'Secure Hash Standard', FIPS PUB 180-1 (Apr. 17. 1995),[7] have been introduced into the key generation process."); id. at col. 1:63-65 ("An example of an irreversible algorithm is the secure hash algorithm as defined by FIPS PUB 180-1, SECURE HASH STANDARD (SHS) Apr. 17, 1995."); '612 Patent, col. 2:2-5.) "Irreversible" is another way of saying that one cannot find the input from the output.

Similarly, the Schneier Book (which is intrinsic evidence) emphasizes that the "computational infeasibility" of determining the input from the output is a requirement of a "secure hash" function by explaining "one-way hash functions have additional characteristics that make them one-way . . . Given h [the output of the function], it is hard to compute M [the input to the function] such that H(M) [the function] = h [the output]." (Schneier Book at 429, Ex. 3; see also Menezes Book at 323 (describing "basic properties and definitions" of cryptographic hash functions, including "for essentially all pre-specified outputs, it is computationally infeasible to find an input which hashes to that output"), Ex. 4.) Thus, a "secure hash" function must make it computationally infeasible to determine the input from the output.

[7] The SHA-1 standard expressly states that "[t]he SHA-1 is called secure because it is computationally infeasible to find a message [i.e., input] which corresponds to a given message digest [i.e., output] . . . ." (FIPS 180-1 at DEFS0006337, Ex. 5.)

(c) The third key property of a "secure hash" function is that it is computationally infeasible to determine two inputs that produce the same output

The third key property that makes a function a "secure hash" function is that it is computationally infeasible to determine two inputs that produce the same output, as the PACid patents confirm. For example, as noted above, the PACid patents identify SHA-1 as not only an example of an algorithm that qualifies as a "secure hash" used in the alleged invention, but as the algorithm "[i]n the preferred embodiment [of the invention]." ('612 Patent, col. 5:1-3; '646 Patent, col. 5:27-28.) The SHA-1 standard, which is intrinsic evidence because it is cited prior art,[8] defines certain attributes of a "secure hash": "The SHA-1 is called secure because it is computationally infeasible to . . . find two different messages [inputs] which produce the same message digest [output]." (FIPS 180-1 at DEFS0006337 (emphasis added), Ex. 5; see also FIPS 180-2 (produced by PACid with its Rule 4-2 disclosures), Ex. 6.)

The Schneier Book (which, again, is part of the intrinsic record) confirms the third requirement when it explains:

> one-way hash functions have additional characteristics that make them one-way . . . Given M [the input to the function], it is hard to find another message M' [another input to the function] such that H(M) [the output for the first input] = H(M') [the output for the second input]

(Schneier Book at 429, Ex. 3; see also id. ("[I]t is hard to find two random messages, M and M' [inputs], such that H(M) [the output for the first input] = H(M') [the output for the second input]."); see also Menezes Book at 323-24 (describing second preimage resistance and collision resistance), Ex. 4.)

Furthermore, the PACid patents embrace the "computationally infeasible" aspect of Defendants' construction because they focus on the feasibility of performing cryptographic

[8] See supra note 4.

analysis in terms of time and computer resources. (See, e.g., '646 Patent, col. 6:18-22 ("[C]ryptographic analysis of the output of a secure hash algorithm is made exceedingly difficult and costly in time and computer resources, since there is no known correlation between the input and the output of the algorithm."); '646 Patent, col. 6:22-23 ("A brute force trial-and-error attack would be even more prohibitive in time and cost.").) Thus, the intrinsic evidence uniformly shows that, for a function to be a "secure hash" function, it must be computationally infeasible to determine two inputs that produce the same output.

(d) The fourth key requirement of a "secure hash" function is that on average approximately 50 percent of it output bits are changed when only one single input bit is changed

Finally, the specification describes a necessary property present in a "secure hash" function: "a secure hash algorithm . . . has the property of changing on average approximately 50 percent of its output bits when only a single bit in the input is changed." ('646 Patent, col. 6:5-9; col. 2:6-8; see also '612 Patent, col. 2:13-15.) This property is sometimes called "strict avalanche criteria." (See Menezes Book at 277 (defining "strict avalanche criterion" as "whenever one input bit is changed, every output bit must change with probability of 1/2"), Ex. 4; Schneier Book at 350 ("Strict avalanche criteria guarantees that exactly half of the output bits change when one input bit changes."), Ex. 3.) An algorithm that meets the first three requirements for a "secure hash" function discussed above will satisfy strict avalanche criteria.

The PACid patents' description of the property of "changing on average approximately 50 percent of the output bits" covers all embodiments of the invention rather than a property limited to only the SHA-1 algorithm identified as the preferred embodiment. Importantly, the patents explain that "a secure hash algorithm" (meaning any secure hash algorithm) as opposed to "the secure hash algorithm" (which might have referred to just one embodiment) has the strict

avalanche criteria property. (See '646 Patent, col. 6:5-9; see also '612 Patent, col. 2:9-15.) As a result, the written description, by virtue of its use of the term in context, defines "secure hash algorithm" as having the property that, if a single bit of the input is changed, on average approximately 50% of the output bits are changed.

Whereas Defendants' constructions capture each of the four key standard requirements for a "secure hash" that the PACid patents embrace, PACid offers constructions, which although unclear, at most, capture only one of these requirements—the requirement that it is computationally infeasible to determine the input from the output. This Court should reject PACid's selective reliance on the specification and adopt Defendants' constructions.

3. "performing a secure hash operation on said shuffled bit result to produce a message digest," "performing a secure hash operation on said first pseudo-random result to . . . produce a second pseudo-random result"

Claim Limitation	Defendants Proposed Construction	PACid Proposed Construction
performing a secure hash operation on said shuffled bit result to produce a message digest ('646 claim 1)	"the input to the secure hash operation is the shuffled bit result from step (a), and the output of the secure hash operation is a message digest"	No separate construction (in light of the others)
performing a secure hash operation on said first pseudo-random result to . . . produce a second pseudo-random result ('612 claim 1)	"the input to the secure hash operation . . . is the first pseudo-random result from step (a) and the output of the secure hash operation is a second pseudo-random result"	No separate construction (in light of the others); alternatively "performing an algorithm that produces a deterministic output having no known relationship with the input that may be used to recover the input from the output"

As an initial matter, PACid asks this Court not to construe these limitations, claiming that the ordinary meaning should control. PACid summarily concludes "no separate construction necessary" for this limitation from the '646 patent because Defendants did not propose a construction for the term "message digest." PACid Br. (Doc. # 251) at 8. Defendants did not identify "message digest" as a term necessary for construction, but rather sought construction of

the entire phrase "performing a secure hash operation on said shuffled bit result to produce a message digest." In any event, the patent specification repeatedly explains that the output of a secure hash operation is a "message digest." Tellingly, PACid provides no explanation as to why it must know the construction of "message digest" in order to propose a construction.

Because PACid has not disclosed its position as to what it believes the plain meaning to be, neither Defendants nor this Court can determine whether the parties have a genuine dispute over these limitations.[9] See, e.g., Maytag Corp. v. Electrolux Home Prods., Inc., 411 F. Supp. 2d 1008, 1036-38 (N.D. Iowa 2006) (failing to provide definition of what the plain meaning is "merely begs the question of what the meaning is"). That PACid does not agree to Defendants' construction is prima facie evidence of a dispute, and absent evidence that the ordinary meaning is undisputed, this Court should address the term. See O2 Micro Int'l Ltd. v. Beyond Innovation Tech. Co., 521 F.3d 1351, 1361 (Fed. Cir. 2008) ("A determination that a claim term 'needs no construction' or has the 'plain and ordinary meaning' may be inadequate when a term has more than one 'ordinary' meaning or when reliance on a term's 'ordinary' meaning does not resolve the parties' dispute.").

Several passages in the PACid patents support Defendants' constructions. First, the disputed claims themselves state that the secure hash operation is "performed on" the "shuffled-bit result" and "pseudo-random result," respectively. These results therefore are inputs to the secure hash operation. The thing "produce[d]"—the output—is the message digest and second pseudo-random result. Thus, this claim language directly leads to Defendants' proposed constructions for these terms.

Figure 2 of the '646 patent provides further support for Defendants' construction:

[9] Although PACid purports to offer an alternative construction for the '612 patent's version of the limitation, this alternative is merely PACid's construction of "secure hash operation" and not the entire limitation at issue.

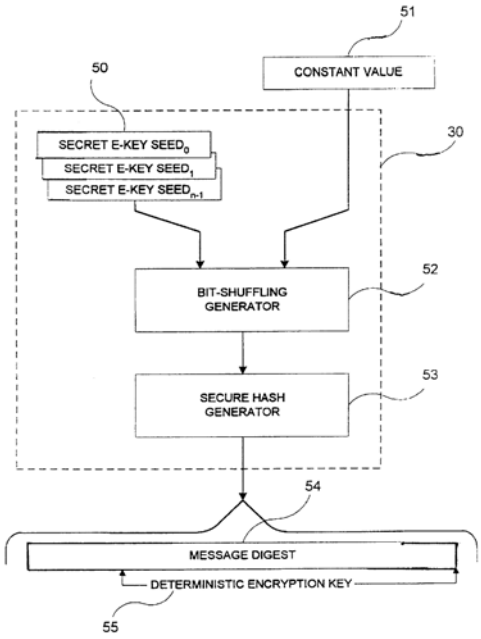

FIG. 2

('646 Patent, Fig. 2.) A downward arrow originates from the bit-shuffling generator and points to the secure hash generator. This arrow shows that the shuffled bit result (the output from the bit-shuffling generator) is the input used for the secure hash operation. An arrow also originates from the secure hash generator and points to the message digest. This second arrow shows that the output of the secure hash generator is the message digest. ('646 Patent, Fig. 2; see also '612 Patent, Fig. 3; col. 1:67–2:18; col. 3:21-24; col. 4:57-67; col. 6:55-59; col. 7:20-22.) Defendants' proposed construction conforms to Figure 2, which is displayed above. Similarly, Figure 6 of the '646 patent discloses a logic flow diagram where the output of the shuffle-bit generator is the input into the bit-shuffling generator, and that the output of the secure-hash generator is the message digest. ('646 Patent, Figure 6.)

Passages throughout the '646 patent likewise explain that the input to the secure hash generator is the output of the bit-shuffling generator, and that the output of the secure hash generator is the message digest. For example:

- More particularly, the bits of a constant value or message are logically, cryptographically and/or algebraically combined with the bits of a secret plural bit sequence (E-Key Seed) to provide a bit-shuffling which results in the mapping of a large number of bits into a first pseudo-random number having fewer bits. The resulting bit sequence then is applied through a secure hash function for increased reversibility. The message digest in turn may be truncated to a desired bit length to provide a repeatable, non-predictable but deterministic, and pseudo-random symmetric encryption key. ('646 Patent, col. 3:22-31 (emphasis added).)

- The bits of the E-Key Seed and the constant value are randomly mixed and mapped to a result C of fewer total bits than the combination of the E-Key Seed and the constant value. The result C is a pseudo-random bit sequence which is applied as an input to a secure hash function generator 53, which in turn produces a second pseudo-random bit sequence referred to as a message digest 54. (Id. at col. 4:65–col. 5:4 (emphasis added).)

- Thereafter, the output of the bit-shuffle program is stored in RAM 112. The CPU next acquires the secure hash program 116 stored in ROM 114 and the bit-shuffle output stored in RAM 112, and executes the secure hash program to produce a message digest. The message digest then is written into the I/O Interface Unit 102 for access by the host system by way of the communication bus 101. The host system may truncate the message digest to produce an encryption key. (Id. at col. 7:14-21 (emphasis added).)

- The E-Key Seed and the constant-value stored in RAM 112 then are combined by a bit-shuffle operation at logic step 158, and the result is applied as an input to a Secure Hash Operation at logic step 159 to produce a message digest. (Id. at col. 8:27-31.)

Like the '646 patent, the '612 patent uniformly supports Defendants' constructions, explaining that the input to the secure hash generator is the pseudo-random result output from the bit-shuffling generator and the output of the secure hash operation is a pseudo-random message digest:

> A constant value or message is logically combined to a secret bit sequence (E-Key Seed) to perform a many-to-few bit mapping which shuffles the bits and provides a pseudo-random result. The result then is applied through a secure hash function generator to perform a second many-to-few bit mapping and provide a pseudo-random message digest.

('612 Patent, col. 3:18-23 (emphasis added).) This passage explains that the output of the shuffle-bit generator is the input to the secure hash function, and that the output of the secure

hash function is a "second many-to-few-bit bit mapping" called "a pseudo-random message digest." Moreover, Figure 3 of the '612 patent provides a graphical illustration supporting the Defendants' construction—with an arrow pointing from the shuffle-bit generator to the secure hash generator, and a second arrow pointing from the secure hash generator to the message digest. (See '612 Patent, Fig. 3; see also '646 Patent, Figs. 2 & 6; col. 3:22-31; col. 4:65–col. 5:5; col. 7:12-20; col. 8:27-31.)

For these reasons, this Court should reject PACid's invitation not to construe these limitations, and instead should adopt Defendants' constructions.

4. "shuffled bit result," "bit shuffling operations," "bit shuffling function," "function to shuffle bits," "bit-shuffle computer program"

Claim Limitation	Defendants Proposed Construction[10]	PACid Proposed Construction
shuffled bit result ('646 claim 1, 3, 18)	"the result of an operation that mixes and maps the bits of its inputs"	"the result of an operation that mixes the bits of its inputs"
bit shuffling operations ('646 claim 18)	"operations that mixes and maps the bits of their inputs"	No separate construction (in light of the others); alternatively: "operations that mix the bits of inputs"
bit shuffling function ('646 claim 19)	"a function that mixes and maps the bits of its inputs"	No separate construction (in light of the others); alternatively: "a function that mixes the bits of its inputs"
function to shuffle bits ('612 claim 1)	"a function that mixes and maps the bits of its inputs"	No separate construction (in light of the others); alternatively: "A function that mixes the bits of its inputs"
bit-shuffle computer program ('646 claim 12)	"computer program that performs a bit shuffle operation (as defined by the court)"	No separate construction (in light of the others); alternatively: "a computer program that mixes the bits of its inputs"

Except for "shuffled bit result," PACid argues that "bit shuffling operations," "bit shuffling function," "function to shuffle bits," and "bit shuffle computer program" do not require

[10] In an effort to narrow the issues before the court, Defendants have removed "randomly" from their constructions.

a separate construction (collectively referred as "shuffle bit terms"). Because the parties dispute the meaning of these terms, however, this Court should provide a construction rather than leave the patents with an uncertain scope. See O2 Micro, 521 F.3d at 1361.

The parties agree that shuffling requires "mixing" but dispute whether "mapping" is also required. "Bit-shuffle" is not a term that has a well-known definition in the field of encryption. Therefore, one must examine the PACid patents' specifications to understand this term. J.T. Eaton & Co. v. Atl. Paste & Glue Co., 106 F.3d 1563, 1568 (Fed. Cir. 1997) (If a term has "no previous meaning to those of ordinary skill in the prior art[,] [i]ts meaning . . . must be found somewhere in the patent."). In this inquiry, one must keep in mind the entire specification of the patent. Phillips, 415 F.3d at 1314.

In the context of the '646 and '612 patents, "bit-shuffling" clearly refers to both mixing and mapping. PACid's construction reciting only "mixing" is (1) contrary to the Summary of the Inventions of both PACid patents, and (2) inconsistent with the definition apparent from the patents' use of the term in context, in particular with reference to the "bit-shuffling" performed by the only disclosed embodiment of the '646 patent and the disclosed embodiments of the '612 patent, all of which require both "mixing" and "mapping."

(a) The Summaries of the Invention Require Both "Mixing" and "Mapping" for the "Shuffle Bit Terms"

Statements made in the Summary of the Invention portion of the specification "more broadly describe the overall invention" of the patent and support a limiting definition of a claim term. See C.R. Bard, Inc. v. United States Surgical Corp., 388 F.3d 858, 864 (Fed. Cir. 2004). The Summary of the Invention for the '646 patent explains that the mapping of bits necessarily occurs when shuffling takes place as excerpted below:

> More particularly, the bits of a constant value or message are logically, cryptographically and/or algebraically combined with the bits of a secret plural bit

> sequence (E-Key Seed) to provide a bit-shuffling which results in the mapping of a large number of bits into a first pseudo-random number having fewer bits.

('646 Patent, col. 3:22-28 (Summary of the Invention) (emphasis added).)

Likewise, the Summary of the Invention in the '612 patent explains that "mapping" occurs when bits are shuffled: "[a] constant value or message is logically combined to a secret bit sequence (E-Key Seed) to perform a many-few-bit mapping which shuffles the bits and provides a pseudo random result." ('612 Patent, col. 3:18-24 (emphasis added).) Shuffling of bits thus requires the "mapping" of bits, which is notably absent from PACid's proposed construction for the "shuffle bit terms."

(b) The Disclosed Embodiments Require Both "Mixing" and "Mapping" for the "Shuffle Bit Terms"

The only disclosed embodiment of the '646 patent requires the "bit shuffling generator" to "mix" the input bits and to "map" the input bits to a result. In the passage below, the disclosed embodiment explains that the input of bits into the bit-shuffling generator is the constant value and the E-Key Seed, and that the shuffling generator will mix these bits and map them to a result C.

> Referring to Fig. 2, an E-Key Seed 50 and constant value 51 are combined by a bit shuffling generator 52 that executes an algebraic, cryptographic and/or logic function, which by way of example but not limitation may be the equation A⊕B=C, where A is the E-Key Seed 50 and B is the constant value 51. The bits of the E-Key Seed and the constant value are randomly mixed and mapped to a result C of fewer total bits than the combination of the E-Key Seed and the constant value.

('646 Patent, col. 4:60–col. 5:1 (describing only disclosed embodiment) (emphasis added).)[11] Likewise, the embodiments of the '612 patent also explain that mixing and mapping occur when a bit is shuffled. ('612 Patent, col. 4:47:53 ("The bits . . . thereby are randomly mixed and mapped from a large binary length to a smaller binary length.").)

[11] Defendants' construction does not exclude the preferred embodiment because under Defendants' construction, algebraic, cryptographic, and/or logic functions (including the equation A⊕B=C) may be used by the shuffling-bit generator, for the purpose of "mix[ing] and mapp[ing] to a result C."

Elsewhere, when describing the only disclosed embodiment, the '646 patent explains that the inputs of the bit-shuffling generator are "subjected to bit-shuffling mapping." ('646 Patent, col. 5:44-48.) This description further emphasizes that "mapping" necessarily occurs when "shuffling" is performed. (Id. ("It is to be understood that the algebraic function executed by the [shuffle] function generator 52, where two inputs which collectively are comprised of a larger number of bits are subjected to bit-shuffling mapping . . . .") (describing only disclosed embodiment).)

In their use of the term in the Summary of the Inventions and in their description of the disclosed embodiments, the PACid patents consistently define "bit-shuffle" to require both mixing and mapping. Accordingly, the Court should adopt Defendants' proposed construction for the "shuffle bit terms."

5. "host system" ('646 claim 12)

Defendants Proposed Construction	PACid Proposed Construction
"computer that inputs command sequences to an encryption key generator"	No construction necessary; alternatively: "a system for providing command sequences"

PACid contends that this Court should not construe the "host system" even though its proposed alternative construction reveals that the term is, in fact, disputed. Because the meaning of "host system" is in dispute, this Court should provide a construction. See O2 Micro, 521 F.3d at 1361.

On the merits, the parties appear to agree that the "host system" inputs command sequences. The dispute centers around PACid's (1) failure to identify the recipient of the command sequences and (2) efforts to rewrite the limitation to delete the word "host" from the '646 patent so that any "system" is claimed rather than the recited "host system."

Addressing the first issue, Defendants' construction for "host system" is rooted in the '646 patent's written description and claims. Claim 12 specifies that an encryption key generator

includes "an I/O interface means in electrical communication with said host system and receiving command sequences from said host system." In addition, the requirement that the command sequences be input to an encryption key generator closely follows the specification, which provides that "a host system . . . inputs commands and data to the key generator system." ('646 Patent, col. 6:63-66; col. 7:50-52 (emphasis added).) Therefore, like Defendants' construction, the claims and the written description are unambiguous that the host system inputs command sequences to an encryption key generator.

Addressing the second issue, PACid's efforts to rewrite the "host system" limitation to delete the word "host" violates the principle that all claim terms are presumed to have meaning in a claim. Innova/Pure Water, Inc. v. Safari Water Filtration Sys., 381 F.3d 1111, 1119 (Fed. Cir. 2004). Deviating from this principle, as PACid does, would be improper here because the intrinsic evidence shows that the "host system" is not merely any "system." The '646 patent explains that the patent is directed to protecting information stored on a computer system or communicated over networks. ('646 Patent, col. 1:41-42.) In this context, a person of ordinary skill would have understood "host" to mean a computer on a network, especially in light of the ordinary meaning of "host." (See, e.g., Am. Heritage Dictionary 849 (4th ed. 2000) (defining "host" as "a computer containing data or programs that another computer can access by means of a network or modem."), Ex. 7; Microsoft Press Computer Dictionary 201 (2d ed. 1994) (defining "host" as "main computer in a system of computers or terminals connected by communications links."), Ex. 8; Webster's New World Dictionary of Computer Terms 240 (6th ed. 1997) (defining "host" as "a server that performs centralized functions, such as making program or data files available to other computers."), Ex. 9.) Thus, the claim's use of the word "host" to modify "system" means that a computer generates the command sequences.

For these reasons, this Court should adopt Defendants' construction.

6. "information file" ('612 claim 1)

Defendants Proposed Construction	PACid Proposed Construction
"a collection of information stored as a unit and identified by a unique name"	"message or file"

The parties' dispute for this limitation pertains to PACid's improper efforts to rewrite the limitation so it recites a term, "message," that the PACid inventors specifically removed during prosecution.

Claim 1 plainly recites an "information file" and not "message or file." During prosecution, the inventors amended the claim to substitute "information file" in place of "message." ('612 Pros. Hist. (12/28/98 Amendment) at 2, 6, Ex. 10.) This change is significant not only because the inventors deliberately chose "information file" during prosecution, see, e.g., Decisioning.com, Inc. v. Federated Dept. Stores, Inc., 527 F.3d 1300, 1309 (Fed. Cir. 2008) (relying on amendments made during prosecution to construe a claim term), but also because it demonstrates the inventors' awareness of the distinction between "information file" and "message" when they made that choice.

Further, the specification for the '612 patent consistently distinguishes between an "information file" and a "message" by referring to the two terms in conjunctive form. ('612 Patent, col. 3:11-13 ("information files and messages").) If the definition for "information file" includes the term "message" as PACid insists, then the term "and messages" in the phrase "information files and messages" in the specification would be redundant and unnecessary.

Moreover, the '612 patent specification does not indicate that the patentee chose to be its own lexicographer or ascribed any special meaning to "information file." For example:

- The encryption key so formed is used to encrypt the information file, and thereafter is destroyed. The encrypted information file and the constant value then are concatenated to place the constant value in the header at the

beginning of the encrypted information file. ('612 Patent, col. 6:60-64 (emphasis added).)

- A method and system is disclosed for protecting sensitive information files and messages from access by unauthorized parties, whether stored in a computer memory or exchanged over a transfer medium between sending and receiving stations." (Id. at col. 3:11-13 (emphasis added).)

Because the '612 patent does not ascribe any special meaning, extrinsic evidence provides insight to what one of ordinary skill in the art would understand the term to mean. A contemporaneous dictionary defines a file as "an organized collection of data that is stored in the external memory of a computer, and can be accessed and manipulated as a single named unit." (Chambers 21st Century Dictionary 489 (1996), Ex. 11.) A person with ordinary skill in the art would understand that an "information file" is a collection of information that does not need to be contiguously stored in memory, but is stored and ultimately accessed as a unit. Defendants' construction adopts this definition.

PACid's construction, in contrast, is flawed because it relies on circular reasoning when using "file" in its definition for "information file." Further, PACid improperly attempts to "read out" the claim limitation "information" from "information file" in order to widen the scope of a claim. Innova/Pure Water, 381 F.3d at 1119-20 (reasoning that all claim terms are presumed to have meaning, the court held that the phrase "operatively connected" took the full breadth of its ordinary meaning, and that the term "operatively" could not just be removed from its construction.).

For the foregoing reasons, this Court should adopt Defendants' construction.

7. "concatenating" ('612 claim 1)

Defendants Proposed Construction	PACid Proposed Construction
"placing one bit field directly next to another"	No construction necessary; alternatively: "linking units together"

As an initial matter, this Court should reject PACid's argument that no construction is necessary because this term is disputed, as the differences between Defendants' construction and PACid's alternative construction demonstrate. See O2 Micro, 521 F.3d at 1361.

The parties' dispute centers on PACid's refusal to adhere to the explicit definition that the '612 patent provides for this term. The '612 patent defines "concatenate" by virtue of its definition of "concatenation": "The term 'concatenation' means that one bit field is juxtaposed to another." ('612 Patent, col. 4:6-7 (emphasis added).)

The specification acts as a dictionary when it expressly defines terms. Martek, 579 F.3d at 1380; CCS Fitness, Inc. v. Brunswick Corp., 288 F.3d 1359, 1366 (Fed. Cir. 2002) ("[A] claim term will not receive its ordinary meaning if the patentee acted as his own lexicographer and clearly set forth a definition of the disputed claim term in . . . the specification . . . ."); Vitronics Corp. v. Conceptronic, Inc., 90 F.3d 1576, 1582 (Fed. Cir. 1996) ("The specification acts as a dictionary when it expressly defines terms used in the claims or when it defines terms by implication."). Because the '612 patent provides an explicit definition of "concatenate," PACid's construction based on a non-technical dictionary is improper.

Defendants' construction "placing one bit field directly next to another," adopts the patent's explicit definition and further clarifies the meaning of the "juxtapose." Dictionaries contemporaneous with the patent's filing show that "juxtapose" means "to place things side by side." (Chambers 21st Century Dictionary 737 (1996), Ex. 11.) When things are placed side by side, those things are placed directly next to another. Therefore, the expressed definition of "concatenation" from the specification, "one bit field [that] is juxtaposed to another," means "placing one bit fit field directly next to another."

This construction is entirely consistent with the specification's description that the constant value is concatenated "at the beginning of the encrypted information file." ('612 Patent, col. 6:61-64 (emphasis added); see also col. 2:67-3:3.) In fact, if the constant value is placed at the beginning of the encrypted information file, then the constant value must be directly next to the encrypted information file. Furthermore, this passage exposes the fatal flaws in PACid's construction because "linking units together" may not necessarily require one unit to be placed at the beginning of another unit.

For these reasons, this Court should adopt Defendants' construction.

8. "algebraic function"[12]

Defendants Proposed Construction	PACid Proposed Construction
"any operation used in mathematics or logic"	No construction necessary; alternatively: "any operation used in mathematics"

PACid and Defendants both propose an alleged "plain meaning" construction for the term "algebraic functions," but the two proposals are potentially quite different. Therefore, this Court should construe this term in order to clarify the proper scope of the claims. See O2 Micro, 521 F.3d at 1361.

In all five claims where this term appears, "algebraic function" is used to limit how the "constant value" and "secret plural bit sequence" are "combin[ed]" either "with an algebraic function to shuffle bits" ('612 patent, Claims 1, 3; '646 Patent, Claims 13, 16) or "to form said shuffled bit result" ('646 patent, Claim 3). In turn, the '646 patent's specification explicitly defines algebraic function in this context:

> It is to be understood that the algebraic function executed by the function generator 52, where two inputs . . . are subjected to a bit-shuffling mapping . . . can be any of numerous other logic, cryptographic, or algebraic functions that would protect the E-Key Seed from being discovered.

[12] "Algebraic function" appears in dependant claims 3, 13, and 16 of the '646 patent, and claim 1 and dependant claim 3 of the '612 patent. The parties agree that this term has the same meaning in all of the asserted claims.

('646 Patent, col. 5:44-50 (emphasis added).) Thus, the patent explains that an "algebraic function" "is to be understood" to include logic functions. This broad definition is consistent with plain meaning of "algebra." Indeed, the term "Boolean algebra" refers to logic functions. (Chambers 21st Century Dictionary 805 (1996) (explaining that computer logic uses "Boolean algebra"), Ex. 11; Webster's New World Dictionary of Computer Terms (6th ed. 1997) at 298 (referring to a "logical operator" as a Boolean operator"), Ex. 9.)

PACid appears to argue that an "algebraic function" can never be a logic function based on claims 1 and 4 of the '612 patent. PACid Br. (Doc. # 251) at 9. Claim 1 refers to "algebraic function" without limitation. Dependent claim 4 narrows claim 1 by stating that a logic function must be included in the step of "combining a constant value and a secret plural bit sequence in accordance with an algebraic function." Thus, if the "algebraic function" of an accused device was arithmetic rather than logic, it would infringe claim 1 but not claim 4. If the "algebraic function" were logic (in other words, Boolean algebra), however, both claims 1 and 4 would be implicated. Nothing about claims 1 and 4 indicate that the term "algebraic function" excludes Boolean algebra / logic functions.

PACid's proposed construction is also problematic because it is unclear whether the word "mathematics" in PACid's construction should be understood broadly enough to include logic functions (one field of mathematics). Rather than leaving ambiguities that lead to future disputes between the parties,[13] Defendants urge this Court to construe "algebraic function" as "any operation used in mathematics or logic."

[13] See, e.g., Fenner Investments, Ltd. v. Microsoft Corp., Case No. 6:07-cv-8-LED (E.D. Tex. Jun. 3, 2009) (Doc. # 325) (ordering a hearing for the day before trial was set to start to address claim construction issues that arose in the context of summary judgment briefing).

9. "logic function"[14]

Defendants Proposed Construction	PACid Proposed Construction
"a function that involves yes-no decisions"	No construction necessary; alternatively: "a function involving operations on variables that may only take a finite number of possible values or states"

This Court should construe "logic function" in order to assist the jury and avoid future disputes about the scope of these claims. In order to narrow issues now at hand, Defendants agree to PACid's alternative proposed construction of this term.

10. "cryptographic function"[15]

Defendants Proposed Construction	PACid Proposed Construction
"a function used in encryption or decryption"	No construction necessary; alternatively: "a function used in encoding or decoding"

This Court should construe "cryptographic function" in order to assist the jury and avoid future disputes about the scope of these claims. The parties initially proposed very similar constructions for this term. Defendants agree with PACid that "cryptographic function" should be construed broadly to encompass any function that can be used in encoding/encrypting or decoding/decrypting. PACid Br. (Doc. # 251) at 15. Based on the clarification in PACid's brief that encoding encompasses encrypting and decoding encompasses decrypting, Defendants can now agree to PACid's alternative proposed construction: "a function used in encoding or decoding."

11. "constant value"[16]

Defendants Proposed Construction	PACid Proposed Construction
"a value that does not change"	No construction necessary; alternatively, "a value that does not change for any given instance of generating an encryption key"

[14] "Logic function" appears in dependant claims 14 and 16 of the '646 patent, and dependant claim 4 of the '612 patent. The parties agree that this term has the same meaning in all of the asserted claims.
[15] "Cryptographic function" appears in dependant claim 15 of the '646 patent, and dependant claim 5 of the '612 patent. The parties agree that this term has the same meaning in all of the asserted claims.
[16] "Constant value" appears in independent claims 1, 3, 6, 12-16 of the '646 patent and claim 1 and 7 of the '612 patent. The parties agree that this term has the same meaning in all of the asserted claims.

PACid argues that no construction is necessary for this term, but provides no reasoning to support this argument. This Court should reject PACid's argument that no construction is necessary because the differences between Defendants' and PACid's alternative constructions show that the parties dispute the meaning of this term. See O2 Micro, 521 F.3d at 1361.

The term "constant value" means "[a] value that does not change." One of ordinary skill in the art would understand that the term "constant" means unchanging, permanent, or fixed. For example, the American Heritage College Dictionary defines "constant" as "[u]nchanging in nature, value, or extent; invariable." (Am. Heritage College Dictionary 298 (3d ed. 1993), Ex. 12.) In all forty-plus instances that the term "constant value" appears, the '646 patent never indicates that the patentee chose to be his own lexicographer to ascribe a meaning to "constant" that deviates from the commonly accepted meaning of this term. The below excerpts provide representative examples of the way the PACid patents use the term "constant value":

- In still another aspect of the invention, the E-Key Seed and constant value may be combined through a sequence of logic, algebraic, and/or cryptographic steps to provide an input to the secure hash function. . . . a further aspect of the invention, the E-Key Seed and constant value may be encrypted to provide an input to the secure hash function. ('646 Patent, col. 3:49:56.)
- The E-Key Seed and the constant value stored in RAM 112 then are combined by a bit-shuffle operation at logic step 158, and the result is applied as an input to a Secure Hash Operation at logic step 159 to produce a message digest. ('646 Patent, col. 8:27-31.)

These excerpts from the '646 patent explain that a constant value may be used as an input to generate a secret key. Likewise, the '612 patent explains "constant value" may be the input to generate a secret key, or that a "constant value" could be a random number or a file name extension:

- Each document or message file is created in normal operation. A constant value or message is logically combined to a secret bit sequence (E-Key Seed) to perform a many-to-few bit mapping which shuffles the bits and provides a pseudo-random result. ('612 Patent, col. 3:16-21 (emphasis added).)

- For example, the constant value 11 may be a random number or file name extension, the operand used in the bit-shuffling function generator 21 could be any algebraic, logical, or encryption operand . . . . ('612 Patent, col. 7:14-16 (emphasis added).)

But in not one instance do the PACid patents teach that a "constant value" is not actually constant, or that a "constant value" fluctuates/varies based on circumstances.

Under PACid's proposed construction, "[a] value that does not change for any given instance of generating an encryption key," a constant value is not really constant. In other words, pi may not always be 3.14159; and gravitational acceleration may not always be 9.8 m/s^2. PACid's reliance on one excerpt from the '612 patent[17] does not support its attempt to deviate from the commonly accepted meaning of "constant" because that passage does not state that the constant value is not constant; nor does it state that the length byte and E-Key Seed ID change the constant value.

PACid's proposed construction assigns no meaning at all to the term "constant." Rather, according to PACid, the constant value needs to remain constant only until a new constant value is formed or, in other words, until the constant value changes. According to PACid, this new formation could occur every second or for every transmission of data. Nothing in the patents contemplates an ever-changing constant value. Accordingly, this Court should adopt Defendants' proposed construction.

[17] PACid relies solely on the '612 patent to construe this term even though the '612 patent and '646 patent are not genealogically related to one another. The passage from the '612 patent relied upon by PACid has no corresponding equivalent in the '646 patent. PACid provides no explanation why this passage from the '612 patent should also apply to the '646 patent.

12. "interrupt control means . . . for issuing an interrupt signal upon receipt of said command sequences" ('646 claim 12)

Defendants Proposed Construction	PACid Proposed Construction
Section 112(6) applies. Function: issuing an interrupt signal upon receipt of said command sequences Corresponding Structure: None; claim is indefinite	Section 112(6) does not apply and no construction necessary; alternatively: "hardware or software that issues a signal to interrupt the operation of a processor" If section 112(6) applies: Function: issuing an interrupt signal upon receipt of command sequences Corresponding Structure: interrupt control unit 104

Defendants incorporate by reference herein their concurrently-filed motion for summary judgment that claims 12 and 26 are indefinite.

IV. CONCLUSION

For the foregoing reasons, Defendants ask this Court to adopt their claim constructions.

Dated: March 5, 2010

Respectfully submitted,

/s/ Jonah D. Mitchell (with permission)
Scott D. Baker (admitted *pro hac vice*)
John P. Bovich (admitted *pro hac vice*)
Jonah D. Mitchell (admitted *pro hac vice*)
James A. Daire (admitted *pro hac vice*)
REED SMITH LLP
Two Embarcadero Center, Suite 2000
San Francisco, CA 94111-3922
Telephone: (415) 543-8700
Facsimile: (415) 391-8269
Email: sbaker@reedsmith.com
Email: jmitchell@reedsmith.com

Rickey L. Faulkner
State Bar No. 06857095
LAW OFFICE OF RICKEY L. FAULKNER, PC
P.O. Box 3367
Longview, TX 75606
Telephone: (903) 248-8246
Facsimile: (903) 248-8249
Email: rick@faulknerlawoffice.com

Counsel for ATHEROS COMMUNICATIONS, INC.

/s/ Yar R. Chaikovsky (with permission)
Yar R. Chaikovsky
John A. Lee
MCDERMOTT WILL & EMERY LLP
275 Middlefield Road, Suite 100
Menlo Park, California 94025
Telephone: (650) 815-7400
Facsimile: (650) 815-7401
Email: ychaikovsky@mwe.com

Counsel for BROADCOM CORPORATION

/s/ Lauren A. Degnan
Ruffin B. Cordell
State Bar No. 04820550
Lauren Degnan (admitted *pro hac vice*)
FISH & RICHARDSON P.C.
1425 K Street NW, 11th Floor
Washington, DC 20005
Telephone: (202) 783-5070
Facsimile: (202) 783-2331
Email: rbc@fr.com
Email: lad@fr.com

Michael E. Jones

State Bar No. 10929400
POTTER MINTON
A Professional Corporation
110 N. College, Suite 500 (75702)
P.O. Box 359
Tyler, TX 75710
Telephone: (903) 597-8311
Facsimile: (903) 593-0846
Email: mikejones@potterminton.com

Eric H. Findlay
State Bar No. 00789886
FINDLAY CRAFT LLP
6760 Old Jackson Hwy, Suite 101
Tyler, TX 75703
Telephone: (903) 534-1100
Facsimile: (903) 534-1137

Counsel for INTEL CORPORATION

/s/ Brian Range (with permission)
James C. Yoon (admitted *pro hac vice*)
Ron E. Shulman (admitted *pro hac vice*)
Brian Range
State Bar No. 24033106
WILSON SONSINI GOODRICH &
ROSATI, Professional Corporation
650 Page Mill Road
Palo Alto, California 94304-1050
Telephone: (650) 493-9300
Facsimile: (650) 565-5100
Email: jyoon@wsgr.com
Email: rshulman@wsgr.com
Email: brange@wsgr.com

Andy Tindel
State Bar No. 20054500
PROVOST UMPHREY LAW FIRM,
L.L.P.
112 East Line Street, Suite 304
Tyler, Texas 75702
Telephone: (903) 596-0900
Facsimile: (903) 596-0909
Email: atindel@andytindel.com

Counsel for MARVELL
SEMICONDUCTOR INC.

CERTIFICATE OF SERVICE

The undersigned hereby certifies that all counsel of record who are deemed to have consented to electronic service are being served with a copy of this document via the court's CM-ECF System per Local Rule CV-5(a)(3) on this 5th day of March, 2010. Any other counsel of record will be served by first class mail on this same date.

/s/ Lauren A. Degnan
Lauren A. Degnan
FISH & RICHARDSON P.C.